**PCT** WORLD INTELLECTUAL PROPERTY ORGANIZATION
International Bureau

INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(51) International Patent Classification 7 : G06F 17/30 — A1

(11) International Publication Number: **WO 00/57302**

(43) International Publication Date: 28 September 2000 (28.09.00)

(21) International Application Number: PCT/US00/04831

(22) International Filing Date: 24 February 2000 (24.02.00)

(30) Priority Data:
09/272,717 19 March 1999 (19.03.99) US

(71) Applicant: ASK JEEVES, INC. [US/US]; 5858 Horton Street, Suite 350, Emeryville, CA 94608 (US).

(72) Inventor: WARTHEN, David; 1455 Washington Avenue, Ablany, CA 94706 (US).

(74) Agents: ALBERT, Philip, H. et al.; Townsend and Townsend and Crew LLP, Two Embarcadero Center, 8th floor, San Francisco, CA 94111 (US).

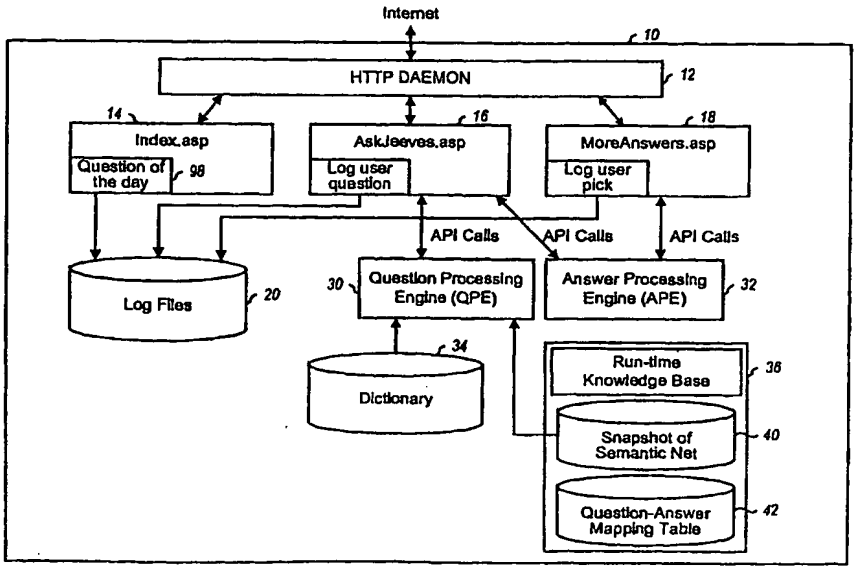
(81) Designated States: AE, AL, AM, AT, AU, AZ, BA, BB, BG, BR, BY, CA, CH, CN, CR, CU, CZ, DE, DK, DM, EE, ES, FI, GB, GD, GE, GH, GM, HR, HU, ID, IL, IN, IS, JP, KE, KG, KP, KR, KZ, LC, LK, LR, LS, LT, LU, LV, MA, MD, MG, MK, MN, MW, MX, NO, NZ, PL, PT, RO, RU, SD, SE, SG, SI, SK, SL, TJ, TM, TR, TT, TZ, UA, UG, UZ, VN, YU, ZA, ZW, ARIPO patent (GH, GM, KE, LS, MW, SD, SL, SZ, TZ, UG, ZW), Eurasian patent (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European patent (AT, BE, CH, CY, DE, DK, ES, FI, FR, GB, GR, IE, IT, LU, MC, NL, PT, SE), OAPI patent (BF, BJ, CF, CG, CI, CM, GA, GN, GW, ML, MR, NE, SN, TD, TG).

**Published**
*With international search report.*
*Before the expiration of the time limit for amending the claims and to be republished in the event of the receipt of amendments.*

(54) Title: GRAMMAR TEMPLATE QUERY SYSTEM

(57) Abstract

An information server directs users of the information server to desired sources of information where the desired sources of information are determined, at least in part, based on user input. The information server includes a query input processor, a question processor and an answer processor. The query input processor is used for accepting an initial user query. The question processor processes the initial user query to identify a set of possible well-formed questions selected from the question database, where a well-formed question is a question in the database that is coupled to at least one answer reference. The answer reference is typically either an answer or a pointer to a possible location of an answer. In a specific embodiment, the information server is coupled to the Internet so that users can pose questions using a Web browser from any Internet-connected device. In some systems, the question processor includes a tokenizer for tokenizing the initial user query into a list of words, a parser for generating a syntactic structure from the list of words, a normalizer for reducing the syntactic structure to a canonical syntactic structure, and a matcher for matching the canonical syntactic structure against a semantic network to obtain a weighted list of well-formed questions representative of possible semantic meanings for the initial user query.



*FOR THE PURPOSES OF INFORMATION ONLY*

Codes used to identify States party to the PCT on the front pages of pamphlets publishing international applications under the PCT.

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| **AL** | Albania | **ES** | Spain | **LS** | Lesotho | **SI** | Slovenia |
| **AM** | Armenia | **FI** | Finland | **LT** | Lithuania | **SK** | Slovakia |
| **AT** | Austria | **FR** | France | **LU** | Luxembourg | **SN** | Senegal |
| **AU** | Australia | **GA** | Gabon | **LV** | Latvia | **SZ** | Swaziland |
| **AZ** | Azerbaijan | **GB** | United Kingdom | **MC** | Monaco | **TD** | Chad |
| **BA** | Bosnia and Herzegovina | **GE** | Georgia | **MD** | Republic of Moldova | **TG** | Togo |
| **BB** | Barbados | **GH** | Ghana | **MG** | Madagascar | **TJ** | Tajikistan |
| **BE** | Belgium | **GN** | Guinea | **MK** | The former Yugoslav Republic of Macedonia | **TM** | Turkmenistan |
| **BF** | Burkina Faso | **GR** | Greece | | | **TR** | Turkey |
| **BG** | Bulgaria | **HU** | Hungary | **ML** | Mali | **TT** | Trinidad and Tobago |
| **BJ** | Benin | **IE** | Ireland | **MN** | Mongolia | **UA** | Ukraine |
| **BR** | Brazil | **IL** | Israel | **MR** | Mauritania | **UG** | Uganda |
| **BY** | Belarus | **IS** | Iceland | **MW** | Malawi | **US** | United States of America |
| **CA** | Canada | **IT** | Italy | **MX** | Mexico | **UZ** | Uzbekistan |
| **CF** | Central African Republic | **JP** | Japan | **NE** | Niger | **VN** | Viet Nam |
| **CG** | Congo | **KE** | Kenya | **NL** | Netherlands | **YU** | Yugoslavia |
| **CH** | Switzerland | **KG** | Kyrgyzstan | **NO** | Norway | **ZW** | Zimbabwe |
| **CI** | Côte d'Ivoire | **KP** | Democratic People's Republic of Korea | **NZ** | New Zealand | | |
| **CM** | Cameroon | | | **PL** | Poland | | |
| **CN** | China | **KR** | Republic of Korea | **PT** | Portugal | | |
| **CU** | Cuba | **KZ** | Kazakstan | **RO** | Romania | | |
| **CZ** | Czech Republic | **LC** | Saint Lucia | **RU** | Russian Federation | | |
| **DE** | Germany | **LI** | Liechtenstein | **SD** | Sudan | | |
| **DK** | Denmark | **LK** | Sri Lanka | **SE** | Sweden | | |
| **EE** | Estonia | **LR** | Liberia | **SG** | Singapore | | |

# GRAMMAR TEMPLATE QUERY SYSTEM

## BACKGROUND OF THE INVENTION

The present invention relates to information query systems in general and to interactive query systems more particularly.

With the increasing popularity of the Internet, the global internetwork of networks, many services have come into being that seek to provide information for users of those services. One such service is the GOPHER service, wherein a user is presented with lists of information in the form of menus, and the user is provided with options to select a menu item, which might lead to another menu or to a document of interest, or to initiate a search with a particular set of keywords. The GOPHER system was primarily text based, with the underlying structure being directories on file servers, usually spanning many interlinked and independent servers.

With the advent of a more graphical interface, many such search services continue today, albeit with a graphical interface for use with a Web browser. Here, "Web" refers to the hyperlinked collection of dynamic and static hypertext pages available over the Internet using the HTTP (HyperText Transport Protocol) and commonly referred to as the "World Wide Web" or "WWW", and a Web browser is a client program which allows a user to navigate the Web.

A typical navigation involves setting up an initial query with a set of search terms and viewing the results. If the results are provided as a hypertext page, the user can then select a link on that hypertext page to view the results in more detail.

A disadvantage of searching using search terms is that the English language is imprecise without context and computers are not good at context. For example, asking a human librarian about "freedom and values in the Victorian era" might yield a book of essays on what the social mores were in the late nineteenth century in Europe and North America. However, performing a search with a search engine against a database of documents might result in the computer returning listings of real estate values for homes with Victorian architecture in Freedom, California.

Many providers of search services have attempted to automate the process of determining the proper context of a query so that the correct meaning is ascribed to

each term. For example, a server might be programmed to note the ambiguity in the term "Victorian" and to further note that "era" refers to a time period and therefore, "Victorian" should be interpreted as the time period. Such processing is quite complex and often still fails to understand the context in which the user asks the question.

## SUMMARY OF THE INVENTION

One embodiment of an information server according to the present invention directs users of the information server to desired sources of information where the desired sources of information are determined, at least in part, based on user input. The information server includes a query input processor, a question processor and an answer processor. The query input processor is used for accepting an initial user query. The question processor processes the initial user query to identify a set of possible well-formed questions selected from the question database, where a well-formed question is a question in the database that is coupled to at least one answer reference. The answer reference is typically either an answer or a pointer to a possible location of an answer.

In a specific embodiment, the information server is coupled to the Internet so that users can pose questions using a Web browser from any Internet-connected device. In some systems, the question processor includes a tokenizer for tokenizing the initial user query into a list of words, a parser for generating a syntactic structure from the list of words, a normalizer for reducing the syntactic structure to a canonical syntactic structure, and a matcher for matching the canonical syntactic structure against a semantic network to obtain a weighted list of well-formed questions representative of possible semantic meanings for the initial user query.

## BRIEF DESCRIPTION OF THE DRAWINGS

Fig. 1 illustrates two variations of an information server according to the present invention; Fig. 1(a) is a block diagram of an HTTP server used to implement the information server and Fig. 1(b) is a block diagram of a generalized server user to implement the information server.

Fig. 2 is an illustration of one possible display for a question entry page.

Fig. 3 is an illustration of one possible display for a question display page; Fig. 3 is an example of a question display page that might result from the processing of the question entry page shown in Fig. 2.

Fig. 4 is an illustration of one possible display for an additional answers page; Fig. 4 is an example of an additional answers page that might result from the processing of the question display page shown in Fig. 3.

Fig. 5 is a block diagram of the QPE and APE of Fig. 1, shown here in further detail.

Fig. 6 is a graph of a portion of a semantic net.

Fig. 7 is an alternate view of a semantic net.

Fig. 8 is a more detailed view of a portion of the semantic net shown in Fig. 7.

Fig. 9 is a block diagram of a computer system for collecting information and generating run-time structures, such as semantic nets, from the collected information.

## DESCRIPTION OF THE SPECIFIC EMBODIMENTS

The following description describes some embodiments of the invention and these examples are not intended to limit the scope of the invention, which is defined by the attached claims. For example, the embodiments shown in the figure assume that the user asks questions of an information server via a Web browser over an Internet connection, but that need not be the case, as the user might access the information server over a network other than the Internet, or might access a local information server without an intervening network.

Fig. 1(a) is a block diagram of one embodiment of an information server 10 that is Internet-based. Information server 10 is an HTTP server, which responds to requests from HTTP clients such as Web browsers. The actual architecture of an Internet HTTP client-server link and HTTP client is not shown, but it should be understood that information server 10 operates on a computer that is designed and configured for Internet traffic, specifically TCP/IP packets encoding HTTP messages and it should be understood that any manner of client computer can be used to operate the Web browser. Since that technology is well-known and readily available, it need not be described further herein.

Information server 10 can either be a dedicated computer, a computing device specifically designed to implement the functions of an information server according to the methods described herein, or information server 10 can be in the form of one or more program code modules designed to run on a general purpose HTTP server as a process of that HTTP server. However implemented, information server 10 usually

comprises the components shown in Fig. 1, such as an HTTP daemon 12 and a basic set of active HTML pages, including an opening page 14, a question page 16 and an additional answers page 18.

Opening page 14 comprises code and text for presenting the user at an HTTP client (such as a Web browser) with an introduction to information server 10. Question page 16 comprises code and text for presenting the user at the HTTP client with a question entry page (such as the page shown in Fig. 2) and a question display page (such as the page shown in Fig. 3). Additional answers page 18 comprises code and text for presenting the user at an HTTP client with additional answers to the questions shown in a question display page. The basic pages are Active Server Page (.asp) programs, but other programming languages could be used. Active Server Pages are one known technique for combining Visual Basic (VB) scripts and HTTP data structures and text. In addition to the basic set of pages, other pages might be provided to HTTP clients, depending on the navigation taken by the user through pages available from information server 10.

As shown in Fig. 1(a), actions taken by users in response to prompts on the basic set of pages are logged in log files 20. Also, as explained below, question page 16 receives questions and presents the questions entered by users to a question processing engine (QPE) 30. In this example, the communication with QPE 30 is via application programming interface (API) calls. As is well-known in the art of API design, the functionality of QPE 30, although shown separately, might be implemented as a set of calls from code in question page 16 to program code compiled into question page 16 (as might be the case with an API link library) with the communication between the two being in the form of function calls, or the program code for implementing QPE 30 could be a separately running process with the communication being in the form of interprocess messages.

Question page 16 receives the data needed to construct a question display page from QPE 30 and from an answer processing engine (APE) 32. Information server 10 also includes a dictionary 34 and a knowledge base 36, which comprises storage for a semantic net snapshot 40 and a question-answer mapping table 42. QPE 30 is coupled to dictionary 34 and semantic net snapshot 40 and uses the information obtained from those sources to generate template questions in response to a user-entered question. Template questions are questions that are mapped to answers in question-answer mapping table 42.

Template questions usually, but not always, include parameters, such as the template question "What is the weather like in <city>?" where "<city>" is a parameter. APE 32 is coupled to knowledge base 36 to obtain information from semantic net snapshot 40 and question-answer mapping table 42.

Fig. 1(b) is a block diagram of a more generalized information server 50. Information server 50 can be implemented in a variety of client-server systems. For example, information server 50 might be built into a kiosk for providing answers to kiosk users' questions, or information server 50 might be connected to a corporate network for answering customer, supplier and/or employee questions. Information server 50 is shown comprising a client interface 60 for providing various pages (70, 72, 74, 76) to a client and interconnections between the various pages and QPE 30 / APE 32.

The circled numbers in Fig. 1(b) indication one possible set of steps that might occur in a session with a user at a client. In step 1, client interface 60 sends an opening page 70 to the client and control is passed to a question entry page 72 (step 2). Client interface 60 presents the client with question entry page 72 (step 3) and receives the user's response (step 4). Information server 50 passes the response to QPE 30, which returns a set of template questions (step 5). Control passes to the code corresponding to a questions display page 74 (step 6), and information server 50 generates the questions display page for presentation to the user. Client interface 60 presents the questions display page to the user (step 7) and waits for the user to select a template question from the questions display page (step 8). The user selects a template question, if more than one is presented, and also selects any necessary parameters. Usually, the parameters will be directly related to the question asked, so the desired parameters might already be selected.

Once the user selects a template question, information server 50 uses AE to generate answers to the questions and either presents the user with one or more URL's of sites that answer the initial question (step 9A) and control passes to an answer display page (step 9B) that presents the user with the answer directly (step 10).

Referring now to Fig. 2, a typical question entry page 80 is shown. When a user is presented with question entry page 80, the user will then enter an initial user query, in an entry box 82, and press a button 84 to begin the process. The user can also view questions being posed by other users in window 86.

The initial user query can be a natural language question (e.g., "Where can I find information on the sport bicycling?") and may well include grammatical errors, or a

set of keywords, such as "info sport bicycling", or a combination of a natural language question and keywords. Keywords might be "noun phrases" such as "King of Spain". When the user presses button 84, the initial user query is sent to information server 50 and client interface 60 passes the query to QPE 30. The query is logged to log files 20 for use in further refining information server 50, as described later in this description.

Typically, the query is in the form of a text string, but in some implementations, the query might be a text string combined with nontext data structures such as a collection of radio button (multiple choice) selections, switches and/or other pointers to selections. It should be understood that many different forms of queries are possible, so long as QPE 30 accepts the form in which the queries are presented. From the initial user query, QPE 30 processes the question to identify a set of template questions. Template questions are questions in a form that allows its context to be easily understood from the question, either because information server 50 has already answered the question to the liking of one or more users, or because an information analyst has input a question-answer template for that question. A question-answer template is a data structure that codes for one or more questions and includes pointers to answers to the coded question. This is explained in more detail in the description below of semantic nets and question-answer mappings (see, for example, Fig. 7). An example of a question answer template is:

**Is it raining in <city>? ==> Pointer to weather page with parameter <city>**

In that example, queries for the weather in many cities is compressed into one question template, with the parameter <city> being a placeholder for the city of interest. In the template, the answer to the question is a pointer to a weather page (such as a page on a weather Web site) with a parameter equal to the city of interest. Thus, using this question-answer template, information server 50 knows how to answer the question "Is it raining in Seattle?", namely by directing the user to a weather server that accepts a city parameter and passing the weather server "Seattle" as the parameter.

Once QPE 30 identifies the set of template questions, those questions are presented to the user as questions display page 74 for selection. Fig. 3 shows an example display 90 resulting from such a questions display page. From that display 90, the user

can select the desired template question and parameters, or can select a button 92 for more answers, resulting in a display such as that shown in Fig. 4.

An information server might also handle incidental displays, which are useful or interesting to users but are not necessarily required for the operation of the system. One example, a "question of the day" generator 98 is shown in Fig. 1(a). Generator 98 might select a question from asked questions to display in a question of the day area 88 on question entry page 80 (see Fig. 2). Other incidental displays might include a real-time question ticker 86 that puts up a random sampling of the questions currently being asked, a display of popular questions in specific categories and the like.

Referring now to Fig. 5, a block diagram of QPE 30 and APE 32 is shown with QPE 30 comprising a tokenizer 150, a parser 155, a normalizer 160 and a matcher 165. Tokenizer 150 converts the initial user query into a list of words and provides the list to parser 155. One structure for conversion is an augmented transition network. Another approach to tokenizing is to scan the initial user query and group words into conceptual strings, removing plurals and suffixes. With such an approach, the longest strings can be grouped first, so that they are given greater priority over shorter strings.

Parser 155 identifies the set of possible syntactic structures that could represent the question(s) being asked and passes the structure set to normalizer 160, with each syntactic structure representing one possible syntactic interpretation of the question. Parser 155 can also deal with adverbs and specialized parsers for ambiguous statements. For example, parser 155 might recognize that the questions "Who is the French president" and "Who is the president of France" has the same underlying syntactical structure.

Normalizer 160 reforms the syntactic structures into canonical forms by replacing synonyms with a canonical term. Using the canonical terms allows for a much more compact set of questions, since the many synonyms do not need to be handled. For example, a user query might use one of the synonyms "drizzle", "storming" or "misting" for raining. By reducing the synonyms to canonical form, the information server does not need to deal with so many questions because, in the above example, four questions collapse into one. Normalizer 160 uses a semantic map, a small portion of which is shown in Fig. 6, to perform the canonical reduction.

Once normalizer 160 has reduced the structure set to a normalized structure set, normalizer 160 passes the normalized structure set to matcher 165. Matcher 165 then matches the normalized structure set against semantic net snapshot 40. The

semantic net is changing as more questions are being asked and the semantic net is being refined, so an information server will often use a snapshot of the state of the semantic net at one point in time. However, either a snapshot or a live copy of the semantic net will work for its purposes in information servers, such as server 10 or server 50. An example of a semantic net is shown in Figs. 7-8.

By matching the normalized structure set against a semantic net, as described in more detail below in the description of Figs. 7-8, matcher 165 obtains a list of instantiated questions (template questions with parameter values identified) and provides those to APE 32. Since the list is of instantiated questions that are based on template questions, they will be found in question-answer mapping table 42 and APE 32 obtains the answers that match the questions.

Fig. 7 illustrates how a semantic net 200 might be organized to be used to map keywords to questions. Once keywords are mapped to questions, the questions are mapped to answers using question-answer mappings 202. A small portion 204 of semantic net 200 is shown in detail in Fig. 8.

Fig. 9 is a block diagram of a computer system for collecting information and generating run-time structures, such as semantic nets, from the collected information.

WHAT IS CLAIMED IS:

1. An information server for directing users of the information server to desired sources of information where the desired sources of information are determined, at least in part, based on user input, the information server comprising:

query input processor for accepting an initial user query;

a question database of template questions, wherein a template question is a question in the database that is coupled to at least one answer reference, the answer reference being one of an answer to the template question or a pointer to a possible location of an answer to the template question;

a question processor that processes the initial user query to identify a set of correlated template questions selected from the question database, wherein the question processor comprises a parser for generating a syntactic structure from the list of words and a normalizer for reducing the syntactic structure to a canonical syntactic structure;

means for presenting at least some of the set of template questions to the user;

means for accepting a user selection of a template question if more than one template question is presented; and

an answer processor which responds to a user selection of a presented template question from the presented set of template questions, wherein the response of the answer processor depends on the at least one answer reference coupled to the user selected question from the presented set of template questions.

2. The information server of claim 1, wherein the user initial query is a text string comprising a sequence of one or more tokens, wherein a token is one or more words that have meaning together and a token is either a natural language question or a keyword string, which is a text string comprising one or more keywords in an order and zero or more logical connectors.

3. The information server of claim 1, wherein the question processor comprises:

a tokenizer for tokenizing the initial user query into a list of words;

a parser for generating a syntactic structure from the list of words;

a normalizer for reducing the syntactic structure to a canonical syntactic structure; and

a matcher for matching the canonical syntactic structure against a semantic network to obtain a weighted list of well-formed questions representative of possible semantic meanings for the initial user query.

4. The information server of claim 3, wherein the matcher comprises:
a structure mapper;
a substructure mapper; and
a keyword mapper.

5. The information server of claim 1, wherein the question processor comprises confidence weighing logic that compares confidence weights for each of the set of correlated template questions selected from the question database and removes template questions from the set that have a confidence weight below a predetermined threshold.

6. The information server of claim 5, wherein the predetermined threshold is a value corresponding to a drop off of confidence weights of template questions in the set.

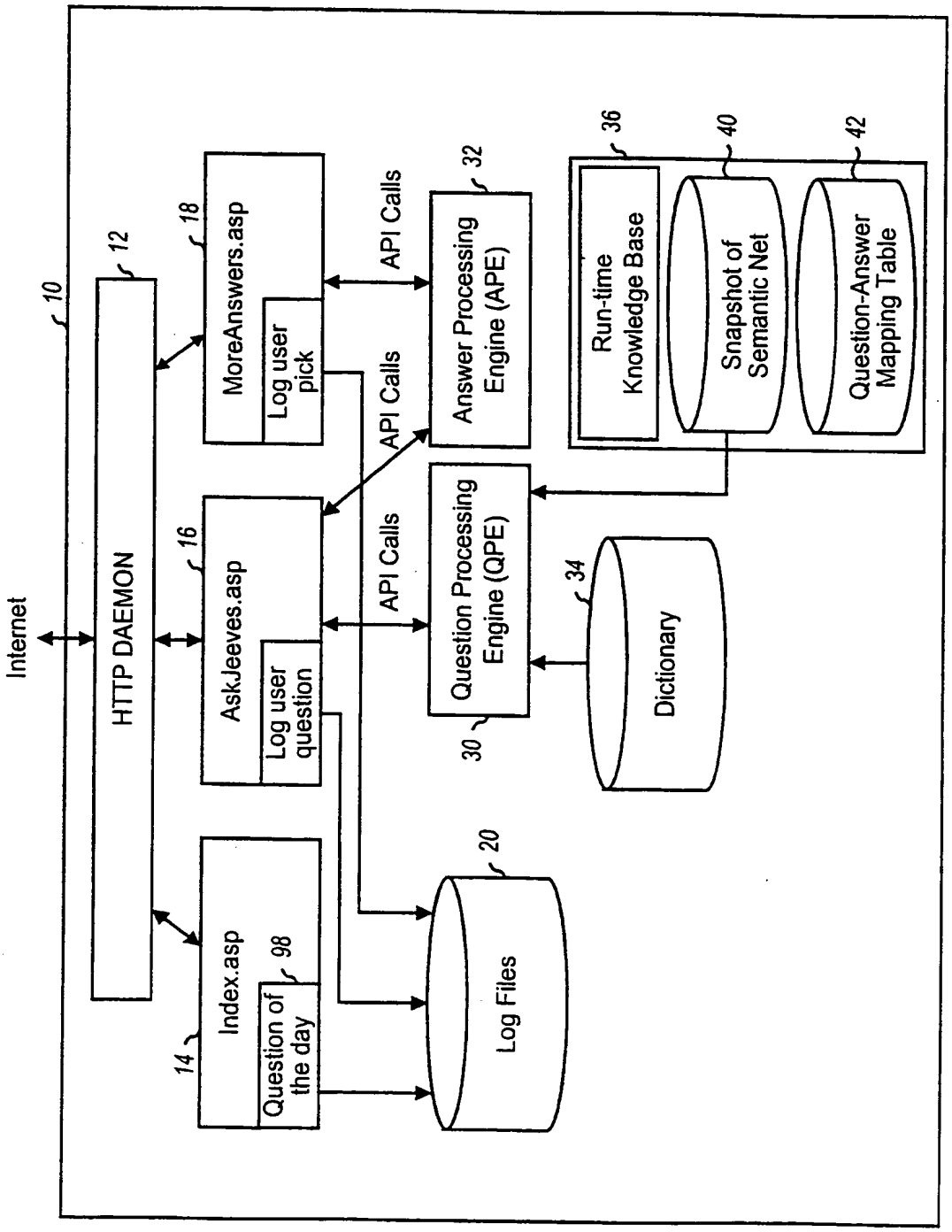

Internet
10
HTTP DAEMON
12
MoreAnswers.asp
18
Log user pick
AskJeeves.asp
16
Log user question
Index.asp
14
Question of the day
98
API Calls
API Calls
API Calls
Answer Processing Engine (APE)
32
Question Processing Engine (QPE)
30
36
Run-time Knowledge Base
40
Snapshot of Semantic Net
42
Question-Answer Mapping Table
34
Dictionary
20
Log Files

FIG. 1(a)

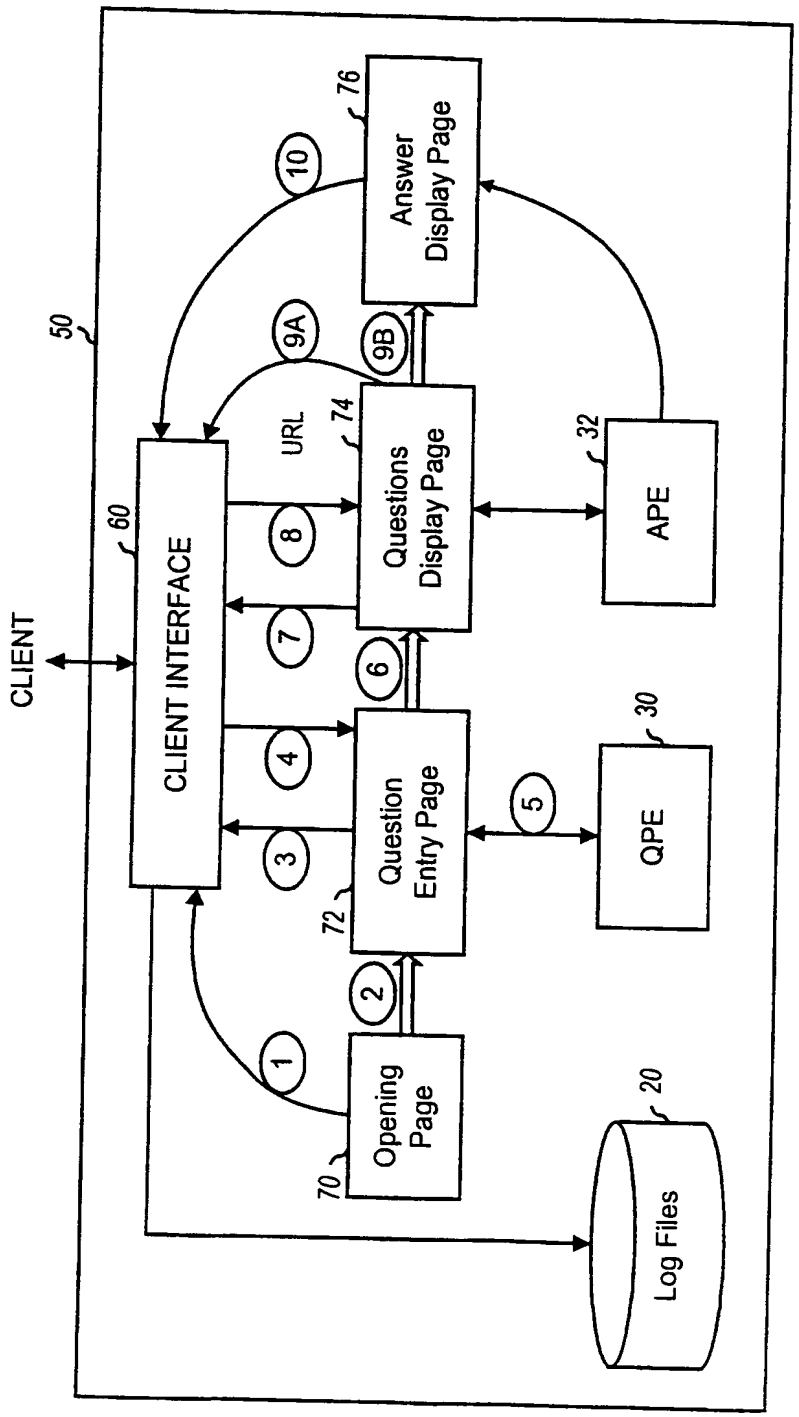

50
CLIENT
CLIENT INTERFACE
60
URL
Answer Display Page
76
Questions Display Page
74
APE
32
Question Entry Page
72
QPE
30
Opening Page
70
Log Files
20
1
2
3
4
5
6
7
8
9A
9B
10

FIG. 1(b)

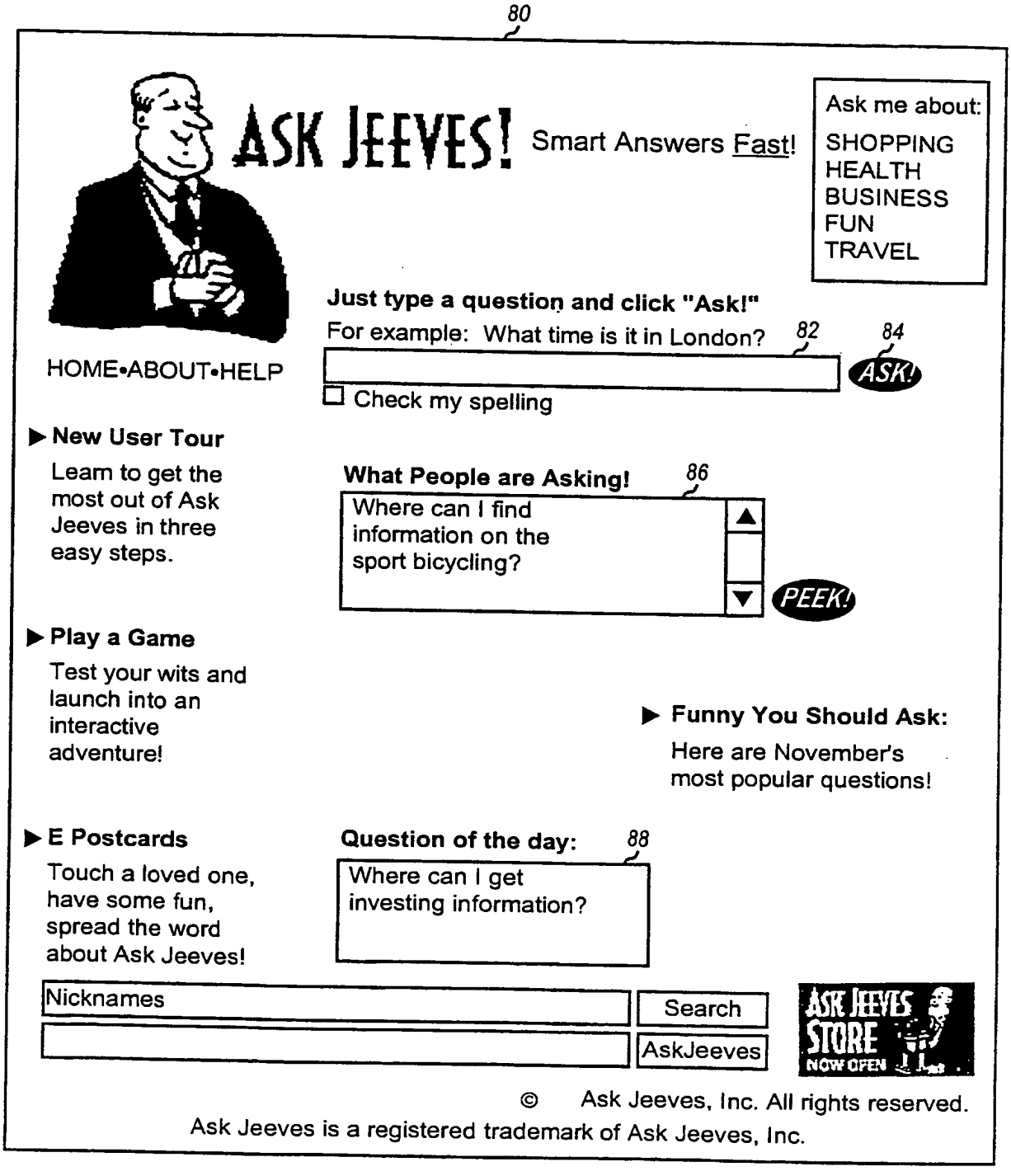

80
ASK JEEVES! Smart Answers Fast!
Ask me about:
SHOPPING
HEALTH
BUSINESS
FUN
TRAVEL
Just type a question and click "Ask!"
For example: What time is it in London?
82
84
ASK!
HOME•ABOUT•HELP
Check my spelling
▶ New User Tour
Learn to get the most out of Ask Jeeves in three easy steps.
What People are Asking!
86
Where can I find information on the sport bicycling?
PEEK!
▶ Play a Game
Test your wits and launch into an interactive adventure!
▶ Funny You Should Ask:
Here are November's most popular questions!
▶ E Postcards
Touch a loved one, have some fun, spread the word about Ask Jeeves!
Question of the day:
88
Where can I get investing information?
Nicknames
Search
AskJeeves
ASK JEEVES STORE NOW OPEN
© Ask Jeeves, Inc. All rights reserved.
Ask Jeeves is a registered trademark of Ask Jeeves, Inc.

FIG. 2

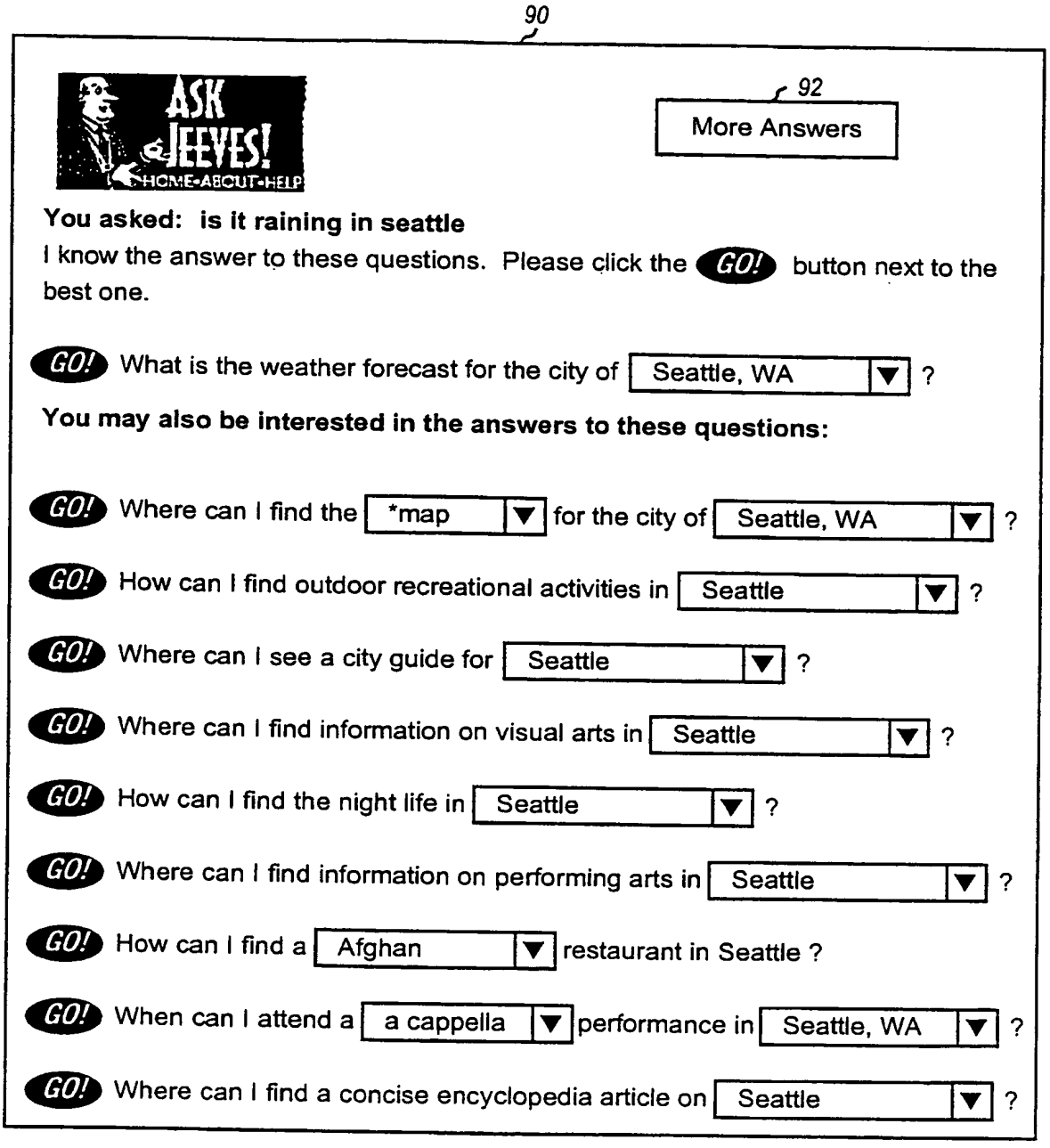

90
ASK JEEVES!
HOME-ABOUT-HELP
92
More Answers
You asked: is it raining in seattle
I know the answer to these questions. Please click the GO! button next to the best one.
GO! What is the weather forecast for the city of Seattle, WA ?
You may also be interested in the answers to these questions:
GO! Where can I find the *map for the city of Seattle, WA ?
GO! How can I find outdoor recreational activities in Seattle ?
GO! Where can I see a city guide for Seattle ?
GO! Where can I find information on visual arts in Seattle ?
GO! How can I find the night life in Seattle ?
GO! Where can I find information on performing arts in Seattle ?
GO! How can I find a Afghan restaurant in Seattle ?
GO! When can I attend a a cappella performance in Seattle, WA ?
GO! Where can I find a concise encyclopedia article on Seattle ?

*FIG. 3*

Ask Jeeves - More Answers

**Jeeves has 2 answers to the question: What is the weather forecast for the city of Seattle, WA?**
Here they are with some brief descriptions.

"WeatherCo." Personal Weather Pages
5 day forecast from "WeatherCo.". 6-10 day forecast and hour-by-hour forecast also available for free, along with various other weather information. Pay subscriptions give you access to aviation info, etc.

American Weather Provider
Weather in American cities.



*FIG. 4*

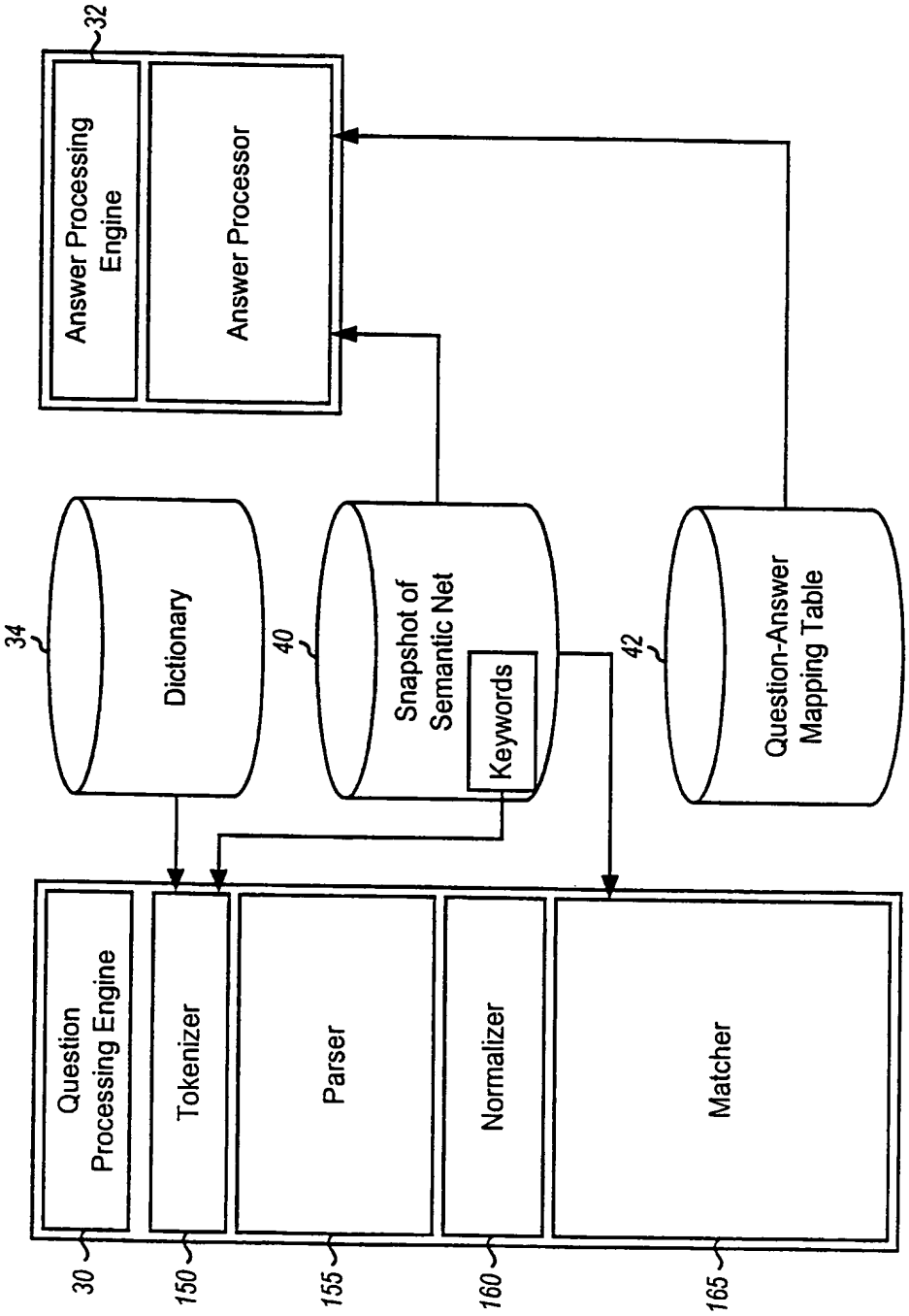

32
Answer Processing Engine
Answer Processor
34
Dictionary
40
Snapshot of Semantic Net
Keywords
42
Question-Answer Mapping Table
Question Processing Engine
Tokenizer
Parser
Normalizer
Matcher
30
150
155
160
165

FIG. 5

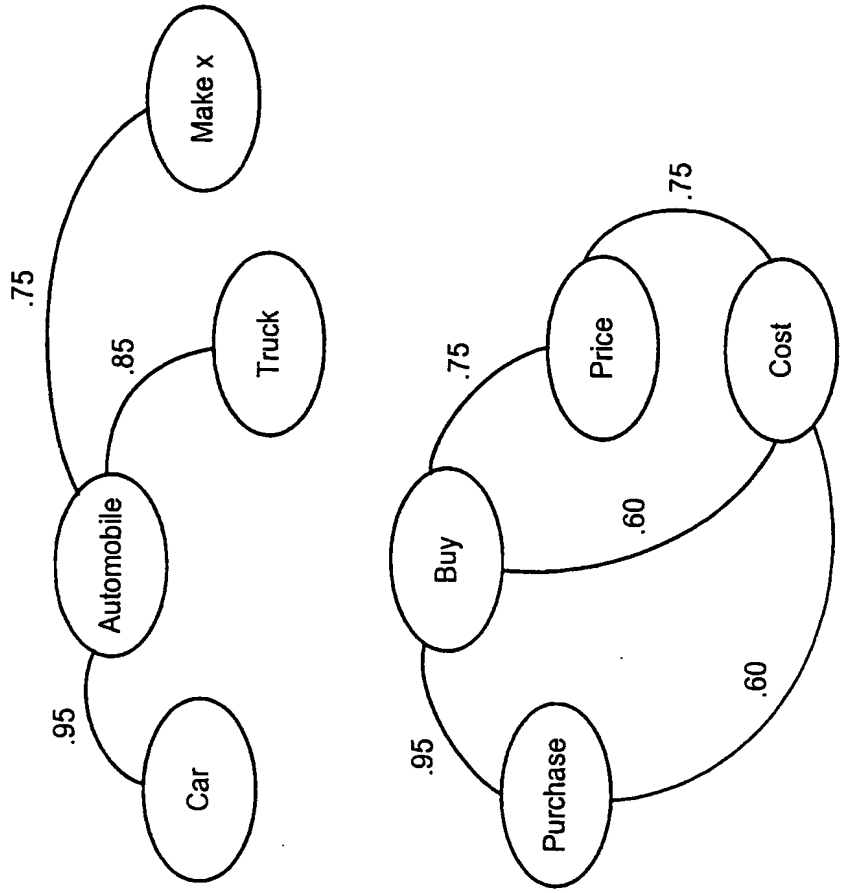

Make x
.75
Truck
.85
Automobile
.95
Car
.75
Price
.75
Cost
Buy
.60
.60
.95
Purchase

FIG. 6

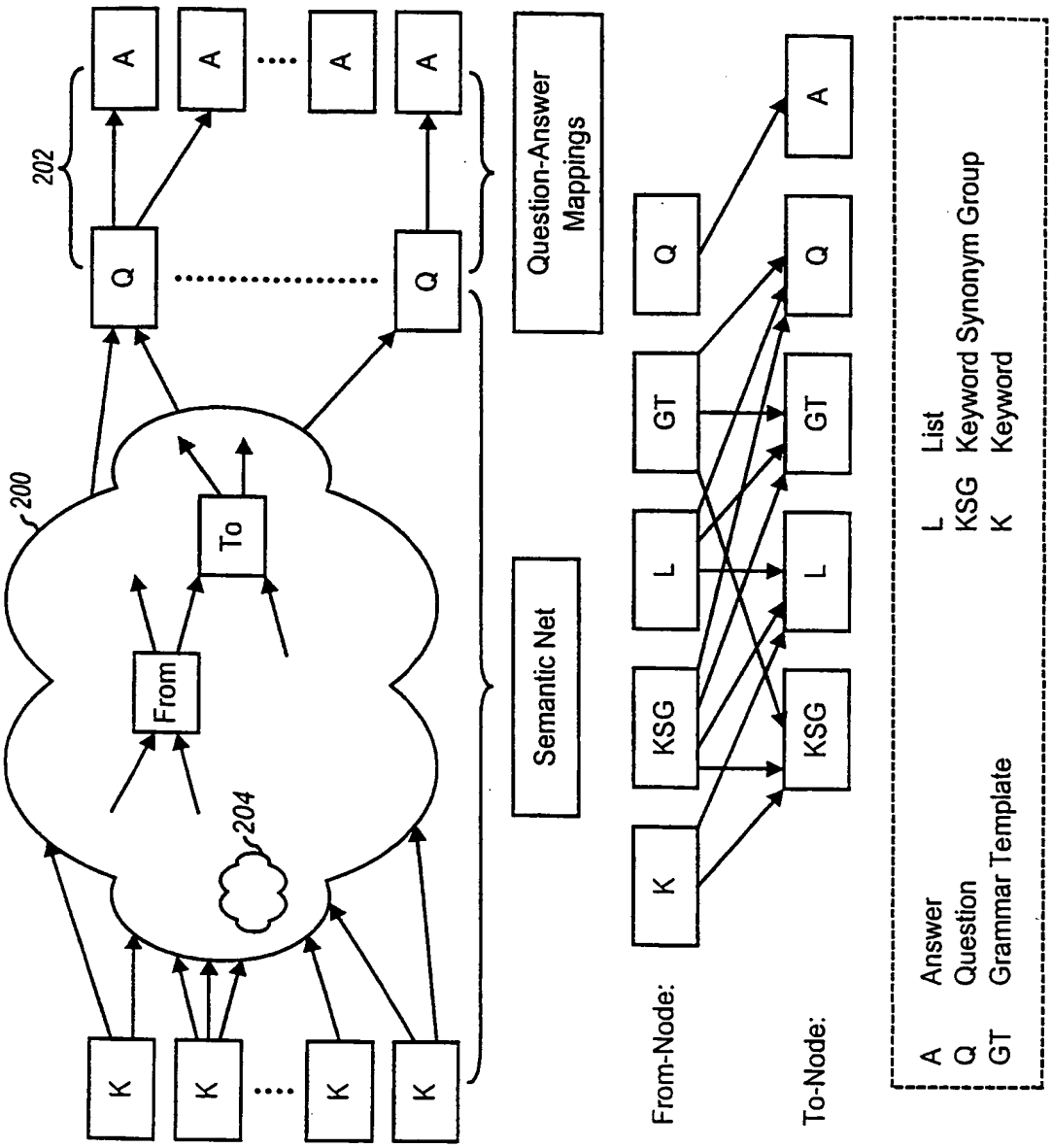

A
Q
GT
Answer
Question
Grammar Template
L List
KSG Keyword Synonym Group
K Keyword
From-Node:
To-Node:
K
KSG
L
GT
Q
A
Question-Answer Mappings
Semantic Net
From
To
200
202
204

FIG. 7

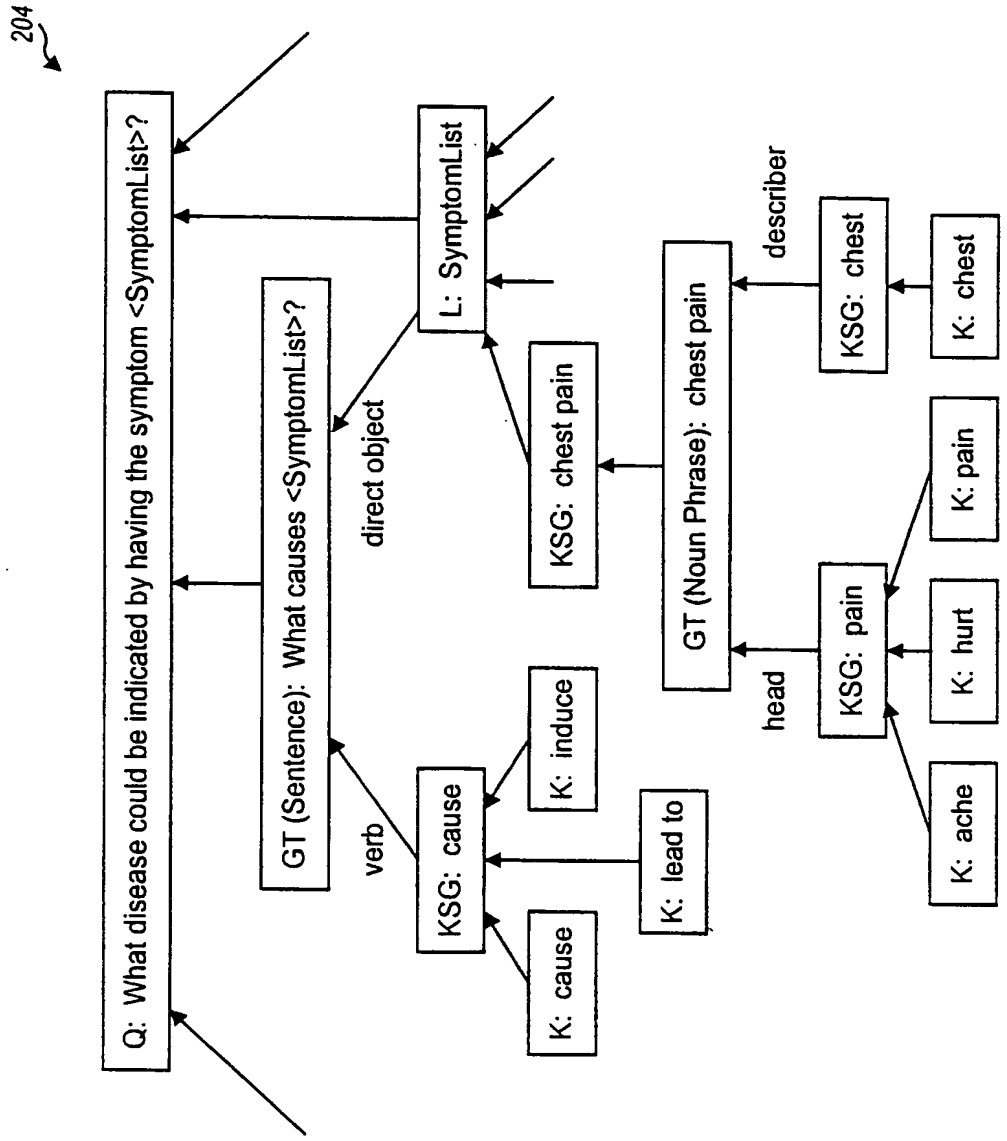

204
Q: What disease could be indicated by having the symptom <SymptomList>?
GT (Sentence): What causes <SymptomList>?
verb
direct object
KSG: cause
K: cause
K: induce
K: lead to
L: SymptomList
KSG: chest pain
GT (Noun Phrase): chest pain
head
describer
KSG: pain
K: ache
K: hurt
K: pain
KSG: chest
K: chest

FIG. 8

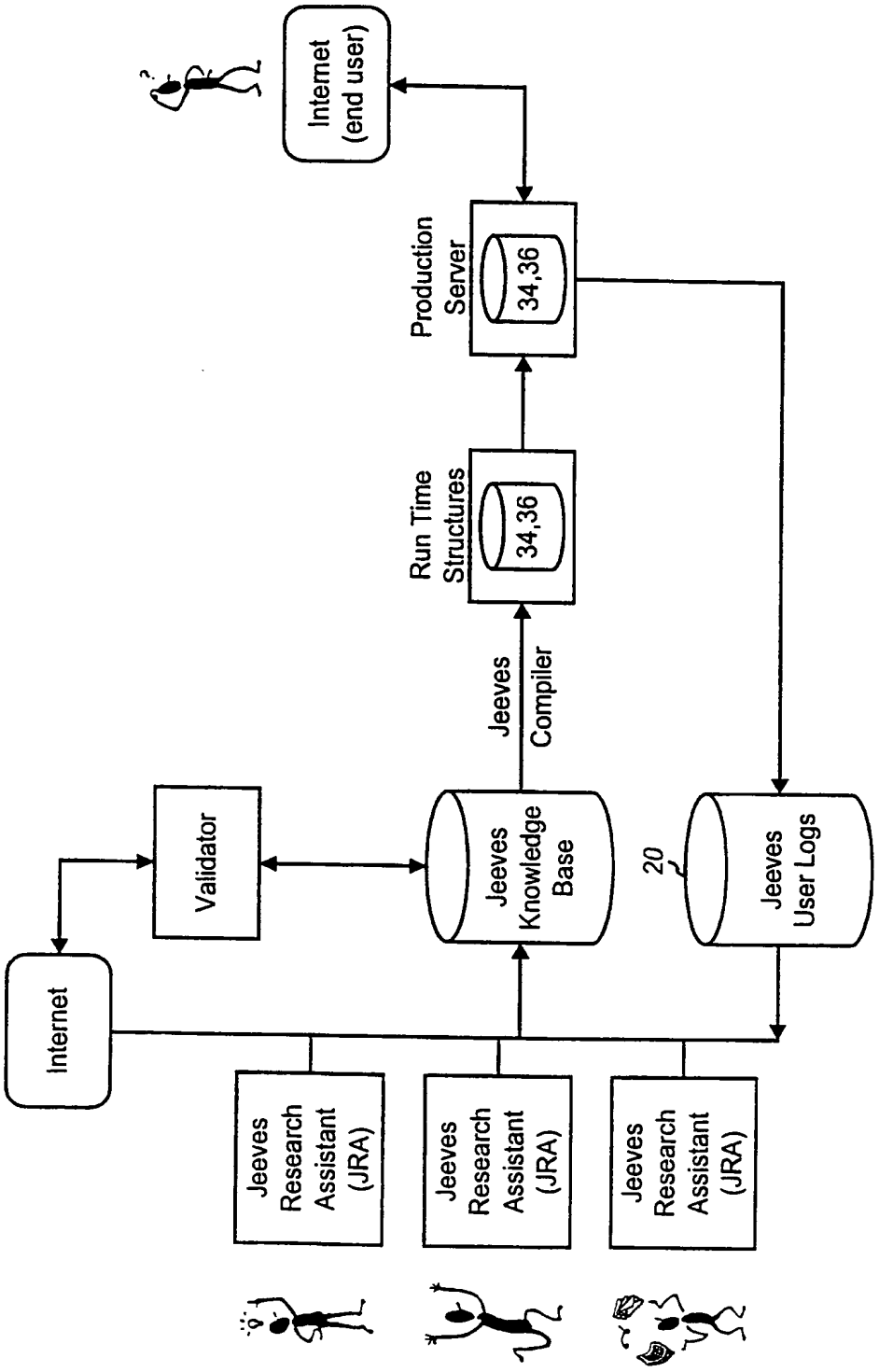

Internet (end user)
Production Server
34,36
Run Time Structures
34,36
Jeeves Compiler
Validator
Jeeves Knowledge Base
20
Jeeves User Logs
Internet
Jeeves Research Assistant (JRA)
Jeeves Research Assistant (JRA)
Jeeves Research Assistant (JRA)

FIG. 9

# INTERNATIONAL SEARCH REPORT

Inter ...n. ...plication No
PCT/US 00/04831

A. CLASSIFICATION OF SUBJECT MATTER
IPC 7 G06F17/30

According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)
IPC 7 G06F

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

Electronic data base consulted during the international search (name of data base and, where practical, search terms used)
EPO-Internal, WPI Data, PAJ, INSPEC, IBM-TDB

C. DOCUMENTS CONSIDERED TO BE RELEVANT

| Category * | Citation of document, with indication, where appropriate, of the relevant passages | Relevant to claim No. |
|---|---|---|
| A | ANICK P G: "INTEGRATING NATURAL LANGUAGE PROCESSING AND INFORMATION RETRIEVAL IN A TROUBLESHOOTING HELP DESK" IEEE EXPERT,US,IEEE INC. NEW YORK, vol. 8, no. 6, 1 December 1993 (1993-12-01), pages 9-17, XP000414491 ISSN: 0885-9000 page 11, left-hand column, line 1 -page 13, left-hand column, line 42; figures 1,2 --- | 1 |
| A | US 5 870 737 A (DOCKTER MICHAEL JON ET AL) 9 February 1999 (1999-02-09) abstract; figure 1 column 3, line 24 -column 4, line 10 --- | 1 |
| | -/-- | |

[X] Further documents are listed in the continuation of box C.

[X] Patent family members are listed in annex.

* Special categories of cited documents :

"A" document defining the general state of the art which is not considered to be of particular relevance

"E" earlier document but published on or after the international filing date

"L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)

"O" document referring to an oral disclosure, use, exhibition or other means

"P" document published prior to the international filing date but later than the priority date claimed

"T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention

"X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone

"Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art.

"&" document member of the same patent family

| Date of the actual completion of the international search | Date of mailing of the international search report |
|---|---|
| 10 July 2000 | 18/07/2000 |

| Name and mailing address of the ISA | Authorized officer |
|---|---|
| European Patent Office, P.B. 5818 Patentlaan 2 NL - 2280 HV Rijswijk Tel. (+31-70) 340-2040, Tx. 31 651 epo nl, Fax: (+31-70) 340-3016 | Deane, E |

Form PCT/ISA/210 (second sheet) (July 1992)

# INTERNATIONAL SEARCH REPORT

Inter ...n. ...plication No
PCT/US 00/04831

C.(Continuation) DOCUMENTS CONSIDERED TO BE RELEVANT

| Category ° | Citation of document, with indication,where appropriate, of the relevant passages | Relevant to claim No. |
|---|---|---|
| A | JAKOBOVITS R. ET AL.: "Implementing Multi-Level Queries in a Database Environment for Vision Research" PROC. SPIE, THE INTERNATIONAL SOCIETY FOR OPTICAL ENGINEERING, vol. 2040, 1995, pages 95-103, XP000923196 USA page 95, line 8 - line 21 page 101, line 19 -page 102, line 23 --- | 1 |
| A | WISDO C.: "Position Paper" AAAI FALL SYMPOSIUM - AI APPLICATIONS IN KNOWLEDGE NAVIGATION AND RETRIEVAL, IRN 0-929280-95-4, 1995, page 154 XP000923310 the whole document --- | 1 |
| A | ANON.: "SQL Statement Library List" IBM TECHNICAL DISCLOSURE BULLETIN., vol. 36, no. 1, January 1993 (1993-01), pages 82-83, XP000333783 IBM CORP. NEW YORK., US ISSN: 0018-8689 the whole document --- | 1 |
| P,X | WO 99 66378 A (SOCRATIX SYSTEMS INC) 23 December 1999 (1999-12-23) abstract; figure 2 page 6, line 12 -page 9, line 12 ----- | 1 |

1

# INTERNATIONAL SEARCH REPORT

Information on patent family members

Inter. application No
PCT/US 00/04831

| Patent document cited in search report | Publication date | Patent family member(s) | Publication date |
|---|---|---|---|
| US 5870737 A | 09-02-1999 | NONE | |
| WO 9966378 A | 23-12-1999 | NONE | |